



Aula 09 - ESP32 - Configuração de Acesso Wifi

Módulo de Internet das Coisas

- Prof.



UNIVERSIDADE
ESTADUAL DO CEARÁ



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO





IA

Objetivos da Aula

- Introduzir a câmera OV2640 do ESP32 TTGO-T
- Demo-Lab 1- Configurar a câmera em uma Wifi existente
- Demo-Lab 2- Configurar IP estático do ESP32
- Demo-Lab 3- Configurar ESP32 como access point
- Demo-Lab 4- Configurar o brilho, resolução, contraste, qualidade da câmera



IA

Câmera OV2640

- Resolução: 2 Megapixel
- Pinagem: 16 pinos (figura ao lado)
- Tipo: regular e fish-eye



Camera		
Y9	—	39
Y8	—	36
Y7	—	23
Y6	—	18
Y5	—	15
Y4	—	4
Y3	—	14
Y2	—	5
VSNC	—	27
HREF	—	25
PCLK	—	19
PWD	—	26
XCLK	—	32
SIOD	—	13
SIOC	—	12
RST	—	NC



SSD1306		
SDA	—	21
SCL	—	22

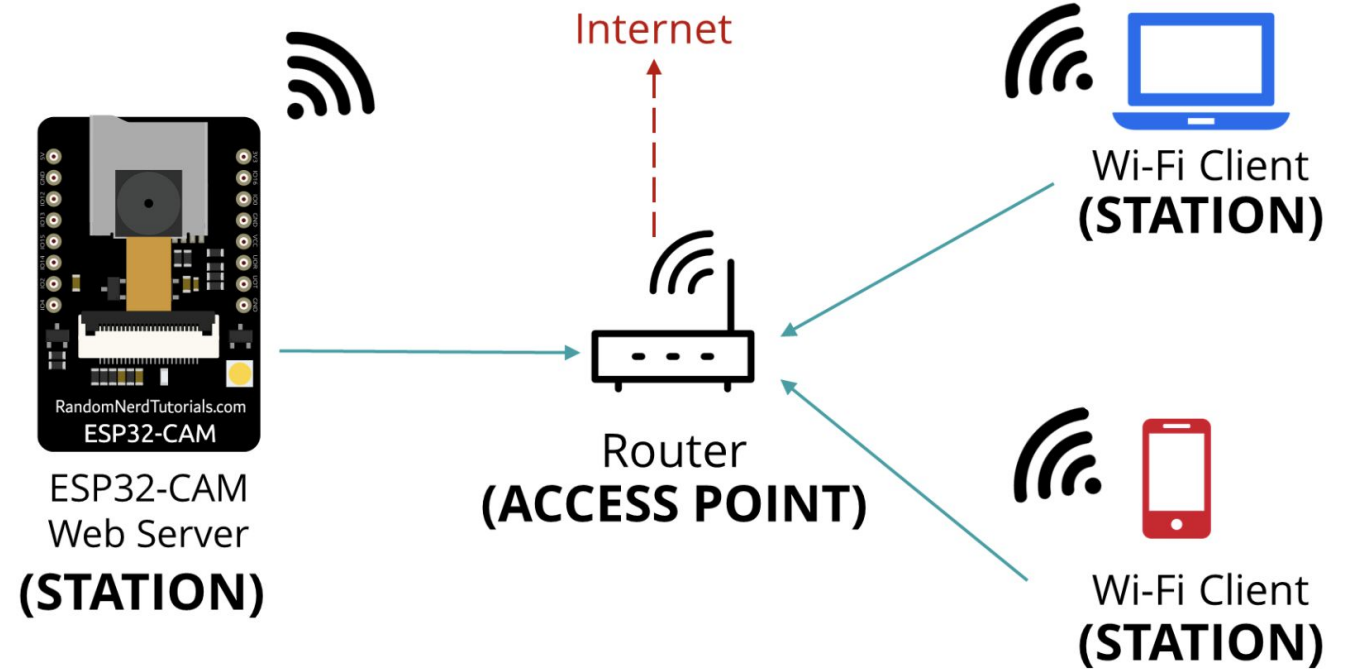
Button		
Left	—	RST
Right	—	34

Pir Sensor		
Pir	—	33



Demo-Lab 1- Configurar a câmera em uma Wifi existente [1]

- Objetivos
 - Acessar o streaming de vídeo da câmera
 - Configurar o ESP32 em uma Wifi existente
 - Executar um código exemplo da ESP32- CAM AI-Thinker
- Passos
 - Usar a Arduino IDE para acessar o código exemplo
 - Fazer o upload do código
 - Obter o endereço IP
 - Acessar o servidor de streaming de vídeo



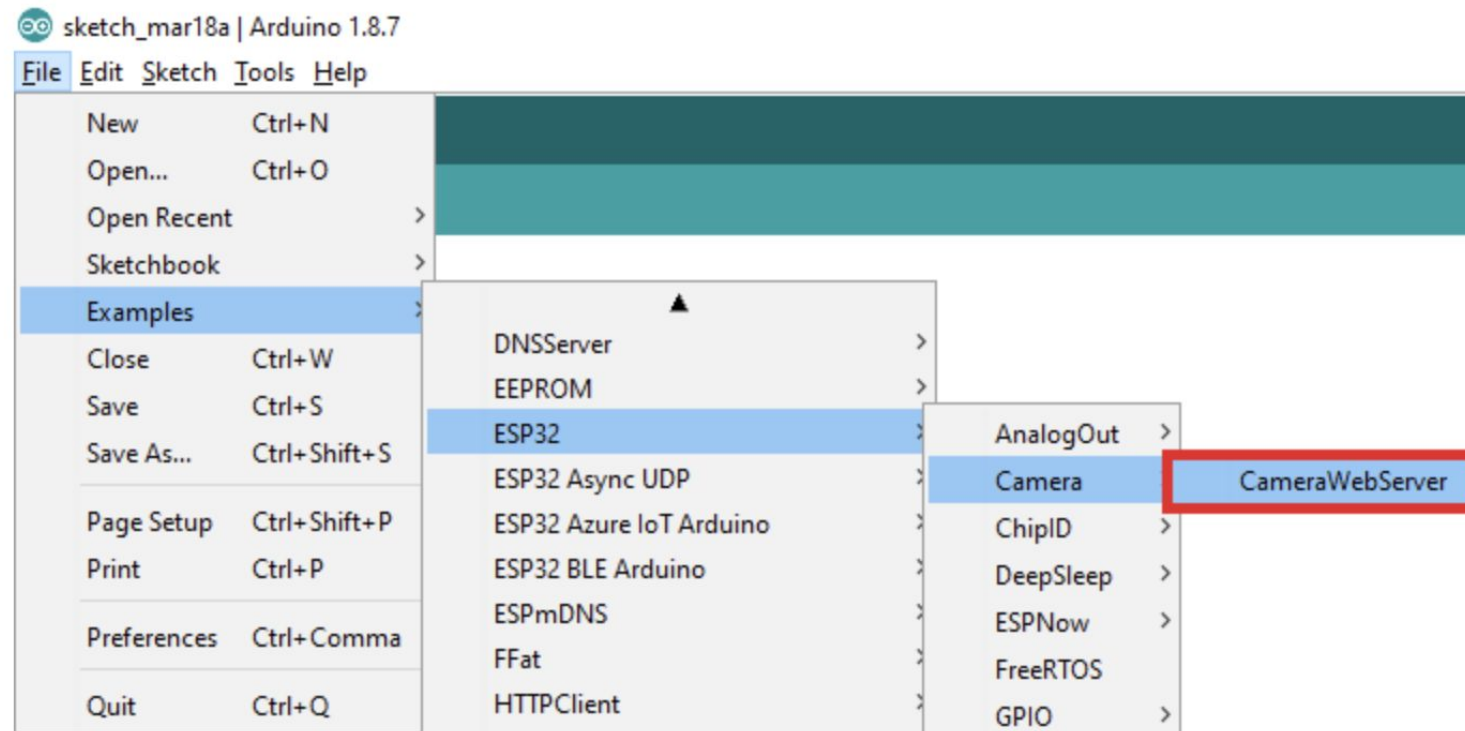


IA

Demo-Lab 1- Configurar a câmera em uma Wifi existente [1]

1 - Código exemplo CameraWebServer

- Na Arduino IDE vá para File>Examples>ESP32>Camera e abra o exemplo CameraWebServer





IA

Demo-Lab 1- Configurar a câmera em uma Wifi existente [1]

- Insira o SSID e senha da rede Wifi no código:
 - SSID: Iracema
 - Passworh: ThinkBig

```
const char* ssid = "REPLACE_WITH_YOUR_SSID";  
const char* password = "REPLACE_WITH_YOUR_PASSWORD";
```



IA

Demo-Lab 1- Configurar a câmera em uma Wifi existente [1]

- Selecione o módulo de câmera correto no código:


```
// Select camera model
//#define CAMERA_MODEL_WROVER_KIT
//#define CAMERA_MODEL_ESP_EYE
//#define CAMERA_MODEL_M5STACK_PSRAM
//#define CAMERA_MODEL_M5STACK_WIDE
#define CAMERA_MODEL_AI_THINKER
```




IA

Demo-Lab 1- Configurar a câmera em uma Wifi existente [1]

3 - Upload do código para o ESP32

- Vá para **Tools>Board** e selecione **AI-Thinker ESP32-Cam**
- Vá para **Tools>Port** e selecione a porta **COM** conectada ao ESP32
- Clique no botão de upload 
- Quando você começar a ver os pontos na janela de debugging pressione o botão RST da placa ESP32

```
esptool.py v2.6-beta1  
Serial port COM10  
Connecting.....
```




IA

Demo-Lab 1- Configurar a câmera em uma Wifi existente [1]

4 - Obtendo um endereço IP

- Abra o Serial Monitor na baud rate 115200. Pressione o botão RST da placa ESP32
- O endereço IP da placa ESP32 será impresso no Serial Monitor

```
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1100
load:0x40078000,len:10088
load:0x40080400,len:6380
entry 0x400806a4

..
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.1.91' to connect
```

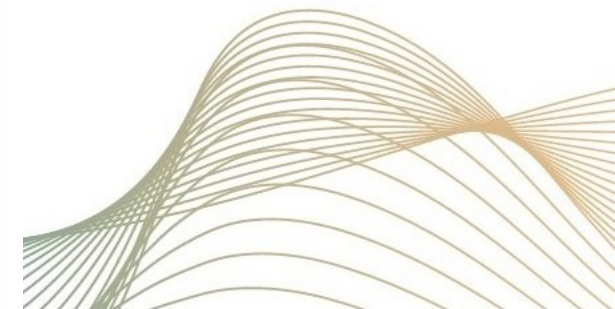
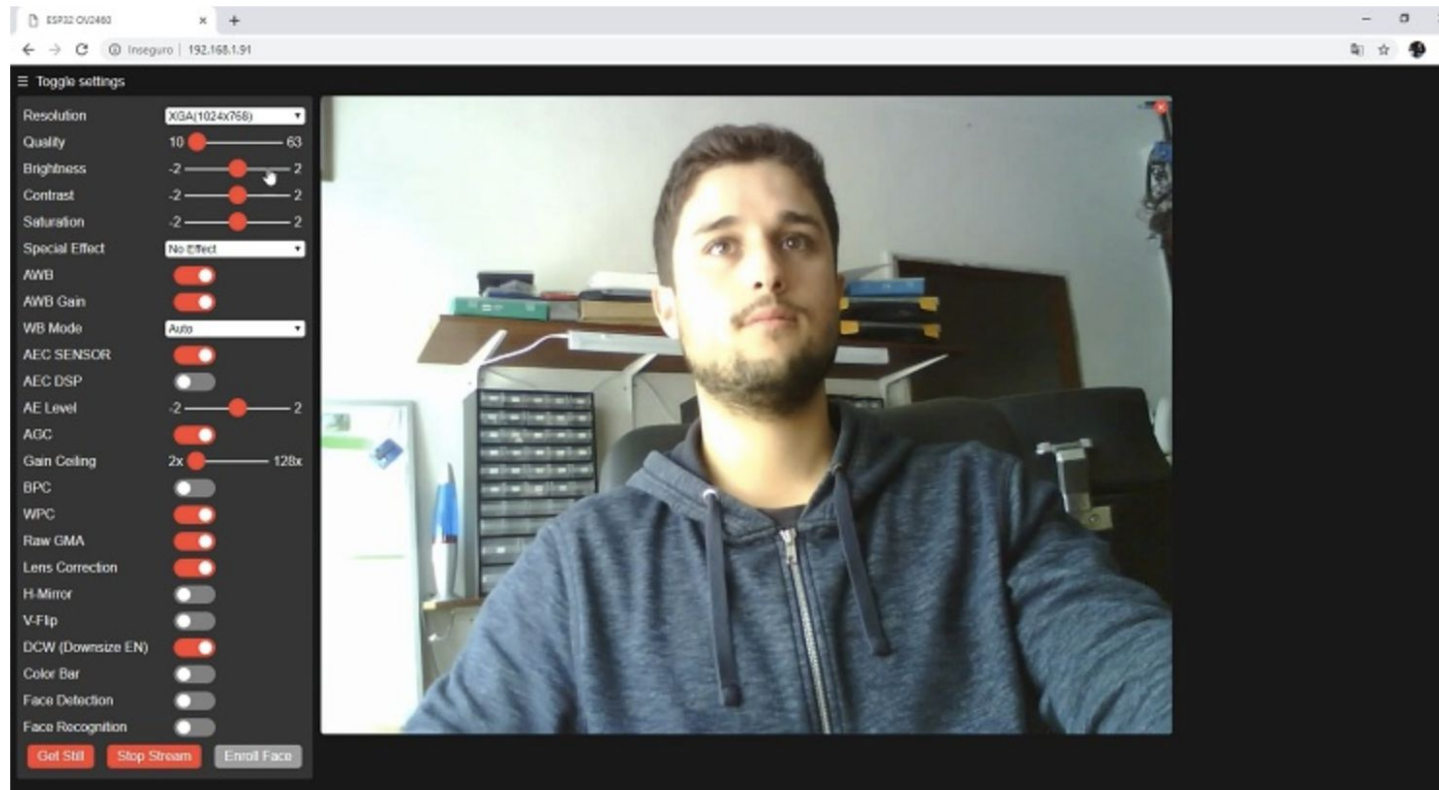


IA

Demo-Lab 1- Configurar a câmera em uma Wifi existente [1]

5 - Acessando o servidor de streaming de vídeo

- Abra um navegador e digite o IP da câmera
- Na página carregada, pressione o botão start streaming





IA

Demo-Lab 2 - Configurar IP estático do ESP32 [2]

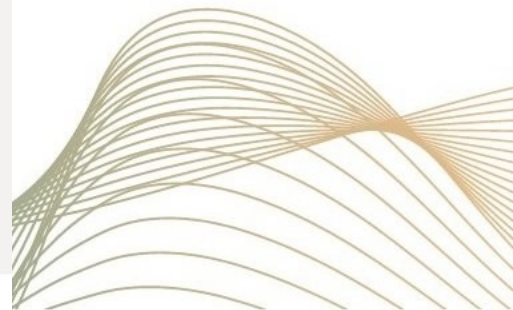
- Objetivos
 - Reservar um endereço IP da Wifi exclusivo para o ESP32
 - Garantir o acesso ao ESP32 pela Wifi após uma reinicialização inesperada
- Passos
 - Modificar o código exemplo do Demo-Lab 1 com endereço IP fixo/estático
 - Compilar e gravar o código
 - Acessar o streaming de vídeo usando IP fixo/estático do ESP32



1 - Configurar o endereço IP estático

- Antes de setup() e loop(), criar as seguintes constantes com informações de endereçamento. Neste exemplo, o endereço IP é 192.168.1.184 e gateway 192.168.1.1

```
// Set your Static IP address  
IPAddress local_IP(192, 168, 1, 184);  
// Set your Gateway IP address  
IPAddress gateway(192, 168, 1, 1);  
IPAddress subnet(255, 255, 0, 0);  
IPAddress primaryDNS(8, 8, 8, 8); //optional  
IPAddress secondaryDNS(8, 8, 4, 4); //optional
```





- No setup(), você precisa chamar o método WiFi.config() para associar as configurações à câmera ESP32. Os parâmetros primaryDNS e secondaryDNS são opcionais e podem ser retirados.

```
if(!WiFi.config(local_IP, gateway, subnet, primaryDNS, secondaryDNS)) {  
    Serial.println("STA Failed to configure");  
}  
WiFi.begin(ssid, password);
```



IA

Demo-Lab 2 - Configurar IP estático do ESP32 [2]

2 - Testar o código

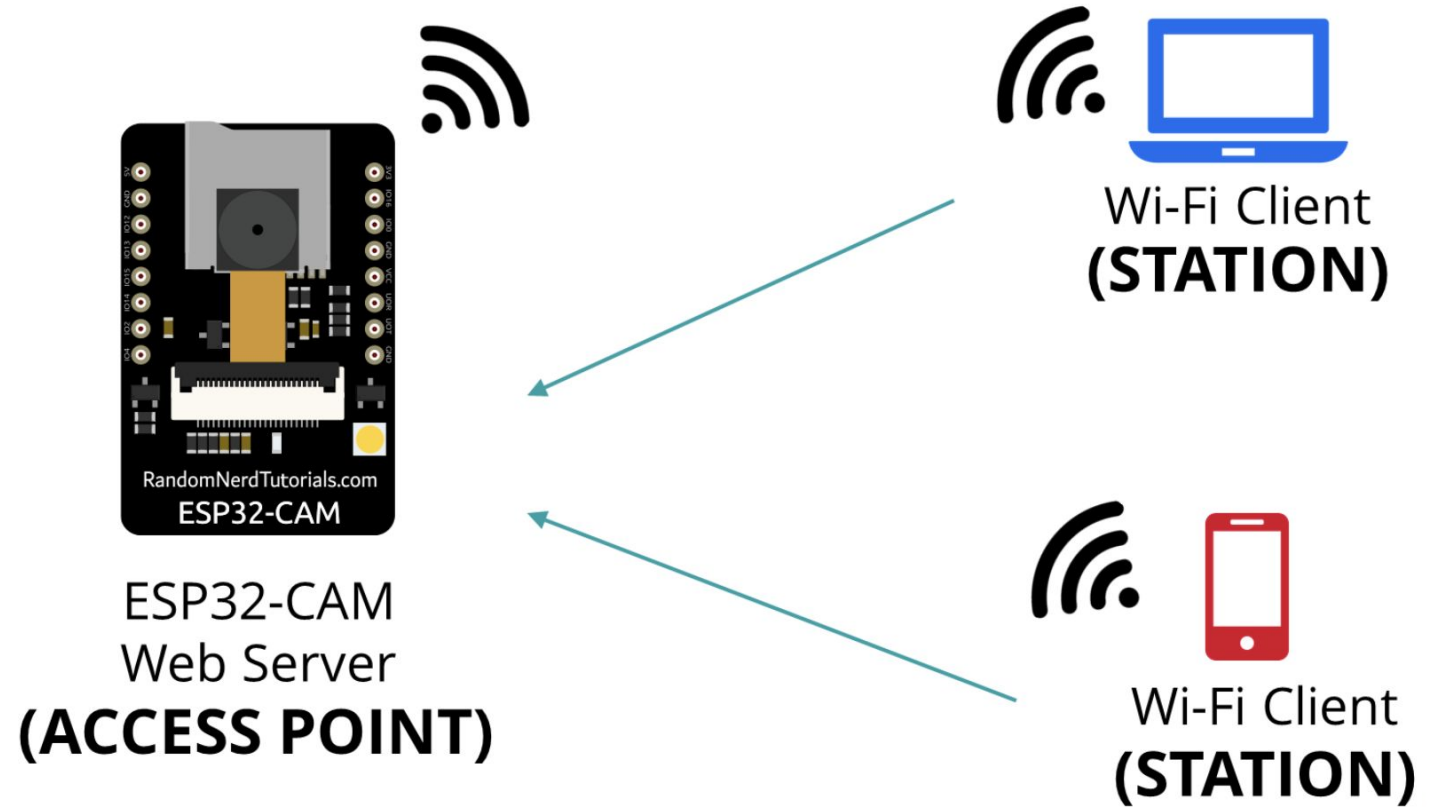
- Depois de fazer o upload do código para placa, abra o Serial Monitor na baud rate 115200
- Reinicie a placa para ver o endereço IP fixo associado.

```
COM3  
|  
Send  
WiFi connected  
Starting web server on port: '80'  
Starting stream server on port: '81'  
Camera Ready! Use 'http://192.168.1.184' to connect  
Autoscroll Show timestamp Newline 115200 baud Clear output
```



Demo-Lab 3 - Configurar o Esp32 como access point [3]

- **Objetivos**
 - Permitir o acesso à câmera quando não houver um AP próximo
 - Configurar o ESP32 como access point criando uma rede Wifi própria
- **Passos**
 - Customizar o SSID e password do código exemplo do Demo Lab 1
 - Configurar o ESP32 como access point
 - Compilar e gravar código
 - Conectar dispositivo na rede Wifi do ESP32
 - Acessar o streaming de video do ESP32





IA

Demo-Lab 3 - Configurar o Esp32 como access point [3]

1 - Customizar SSID e senha

- Você precisa criar um SSID e senha para o ESP32. Por exemplo:

```
const char* ssid = "ESP32-CAM Access Point";  
const char* password = "123456789";
```




2 - Configurar o ESP32 como access point

- Na setup(), remover as seguintes linhas:

```
WiFi.begin(ssid, password);  
while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    Serial.print(".");  
}  
Serial.println("");  
Serial.println("WiFi connected");
```



IA

Demo-Lab 3 - Configurar o Esp32 como access point [3]

- Na setup (), adicionar as seguintes linhas:

```
WiFi.softAP(ssid, password);
```



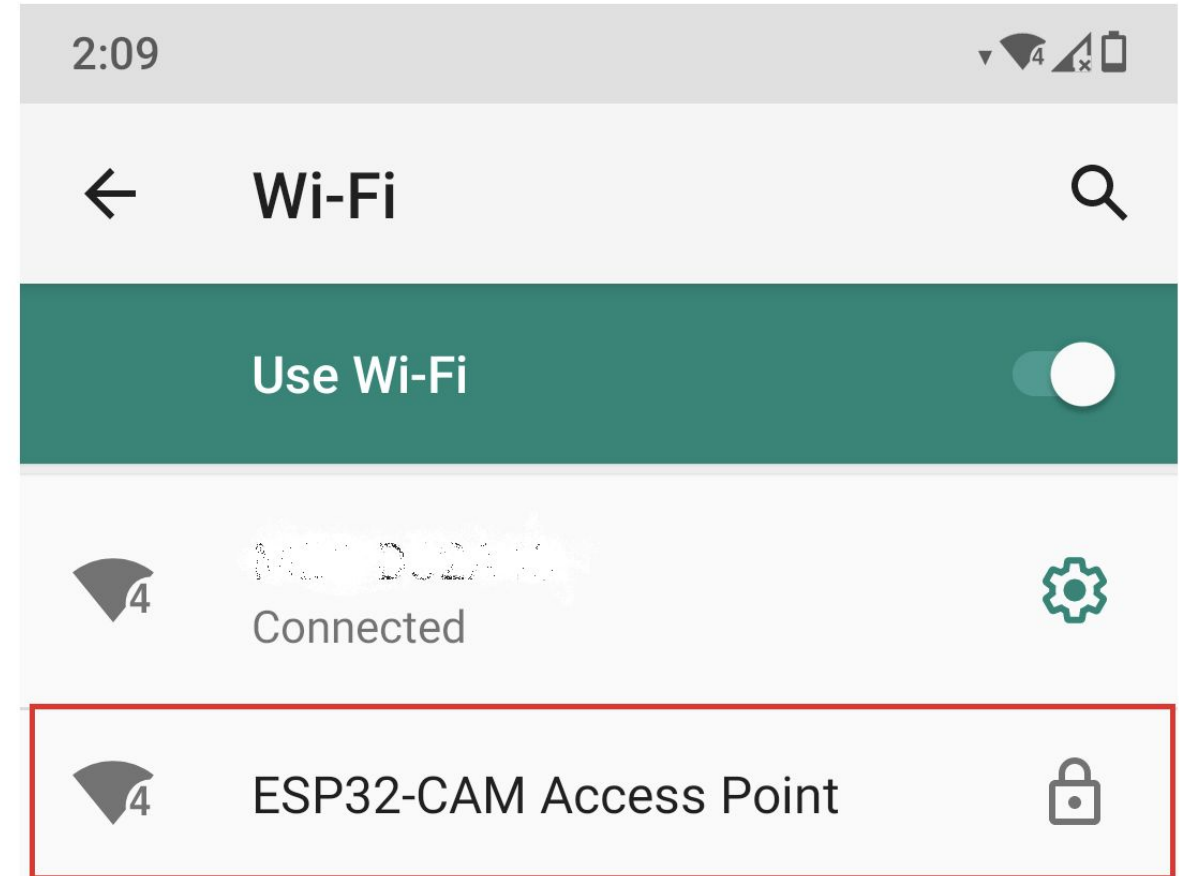
IA

Demo-Lab 3 - Configurar o Esp32 como access point [3]

3. Conectar à rede do ESP32

- Depois de fazer o upload do código para a placa, procure a rede Wifi do ESP32 para conectar-se

4 - Abra o navegador e digite o endereço IP 192.168.4.1 para acessar o servido de streaming de vídeo





IA

Demo-Lab 4 - Configurar brilho, etc. da câmera [4]

- Objetivos
 - Configurar as características da câmera: contraste, brilho, resolução etc.
- Passos
 - Modificar o código exemplo do Demo-Lab 1
 - Adicionar código de configuração depois daquele que inicializa a câmera



The image shows a close-up of an ESP32-CAM module, a small electronic board with a camera lens and various components. The text 'ESP32-S AI-Thinker' and 'FCC ID: 2A4HMR-ESP32S' is visible on the board.

Set ESP32-CAM Settings

- Contrast
- Brightness
- Resolution
- Quality
- Saturation
- Special Effects
- White Balance
- Mirror and Flip
- Exposure
- Etc...



1 - Mudar as configurações da câmera pelo Sketch Arduino

- Depois de inicializar a câmera, adicione essas linhas:

```
sensor_t * s = esp_camera_sensor_get();

s->set_brightness(s, 0);      // -2 to 2
s->set_contrast(s, 0);        // -2 to 2
s->set_saturation(s, 0);      // -2 to 2
s->set_special_effect(s, 0);  // 0 to 6 (0 - No Effect, 1 - Negative, 2 -
s->set_whitebal(s, 1);        // 0 = disable , 1 = enable
s->set_awb_gain(s, 1);         // 0 = disable , 1 = enable
s->set_wb_mode(s, 0);         // 0 to 4 - if awb_gain enabled (0 - Auto,
s->set_exposure_ctrl(s, 1);   // 0 = disable , 1 = enable
s->set_aec2(s, 0);            // 0 = disable , 1 = enable
s->set_ae_level(s, 0);        // -2 to 2
s->set_aec_value(s, 300);     // 0 to 1200
s->set_gain_ctrl(s, 1);       // 0 = disable , 1 = enable
s->set_agc_gain(s, 0);        // 0 to 30
s->set_gainceiling(s, (gainceiling_t)0); // 0 to 6
s->set_bpc(s, 0);            // 0 = disable , 1 = enable
s->set_wpc(s, 1);            // 0 = disable , 1 = enable
s->set_raw_gma(s, 1);        // 0 = disable , 1 = enable
s->set_lenc(s, 1);           // 0 = disable , 1 = enable
s->set_hmirror(s, 0);        // 0 = disable , 1 = enable
s->set_vflip(s, 0);          // 0 = disable , 1 = enable
s->set_dcw(s, 1);            // 0 = disable , 1 = enable
s->set_colorbar(s, 0);       // 0 = disable , 1 = enable
```



2 - Modificar os pinos no começo do código de CameraWebServer

```
#define PWDN_GPIO_NUM      26
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      32
#define SIOD_GPIO_NUM      13
#define SIOC_GPIO_NUM      12

#define Y9_GPIO_NUM        39
#define Y8_GPIO_NUM        36
#define Y7_GPIO_NUM        23
#define Y6_GPIO_NUM        18
#define Y5_GPIO_NUM        15
#define Y4_GPIO_NUM         4
#define Y3_GPIO_NUM        14
#define Y2_GPIO_NUM         5
#define VSYNC_GPIO_NUM     27
#define HREF_GPIO_NUM      25
#define PCLK_GPIO_NUM      19

#define LED_GPIO_NUM       33
```



IA

Demo-Lab 4 - Configurar brilho, etc. da câmera [4]

3 - Configurar a frequência da câmera

```
config.xclk_freq_hz = 20000000;
```

4 - Configurar formato da imagem

```
config.pixel_format = PIXFORMAT_JPEG; //YUV422, GRAYSCALE, RGB565, JPEG
```

o formato da imagem pode ser:

- PIXFORMAT_YUV422
- PIXFORMAT_GRAYSCALE
- PIXFORMAT_RGB565
- PIXFORMAT_JPEG



IA

Demo-Lab 4 - Configurar brilho, etc. da câmera [4]

5 - Configurar o tamanho do frame

```
// Select lower framesize if the camera doesn't support PSRAM
if(psramFound()){
    config.frame_size = FRAME_SIZE_UXGA; // FRAME_SIZE_ + QVGA|CIF|VGA|SVGA|
    config.jpeg_quality = 10; //10-63 lower number means higher quality
    config.fb_count = 2;
} else {
    config.frame_size = FRAME_SIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}
```




6 - Inicializar a câmera

```
// Initialize the Camera
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
}
```



7 - Configurações da OV2640: mudam a imagem

```
sensor_t * s = esp_camera_sensor_get();
s->set_brightness(s, 0);      // -2 to 2
s->set_contrast(s, 0);        // -2 to 2
s->set_saturation(s, 0);       // -2 to 2
s->set_special_effect(s, 0);   // 0 to 6 (0 - No Effect, 1 - Negative, 2
s->set_whitebal(s, 1);         // 0 = disable , 1 = enable
s->set_awb_gain(s, 1);          // 0 = disable , 1 = enable
s->set_wb_mode(s, 0);           // 0 to 4 - if awb_gain enabled (0 - Auto,
s->set_exposure_ctrl(s, 1);    // 0 = disable , 1 = enable
s->set_aec2(s, 0);              // 0 = disable , 1 = enable
s->set_ae_level(s, 0);          // -2 to 2
s->set_aec_value(s, 300);       // 0 to 1200
s->set_gain_ctrl(s, 1);         // 0 = disable , 1 = enable
s->set_agc_gain(s, 0);          // 0 to 30
s->set_gainceiling(s, (gainceiling_t)0); // 0 to 6
s->set_bpc(s, 0);               // 0 = disable , 1 = enable
s->set_wpc(s, 1);               // 0 = disable , 1 = enable
s->set_raw_gma(s, 1);           // 0 = disable , 1 = enable
s->set_lenc(s, 1);              // 0 = disable , 1 = enable
s->set_hmirror(s, 0);           // 0 = disable , 1 = enable
s->set_vflip(s, 0);             // 0 = disable , 1 = enable
s->set_dcw(s, 1);               // 0 = disable , 1 = enable
s->set_colorbar(s, 0);          // 0 = disable , 1 = enable
```



- [1] Rui Santos. Streaming de Video com câmera do ESP32 e reconhecimento facial. 2019. Disponível em: <https://randomnerdtutorials.com/esp32-cam-video-streaming-face-recognition-arduino-ide/>>. Acesso em: 03/04/2023.
- [2] Rui Santos. Configurar o endereço IP estático no ESP32. Disponível em: <https://randomnerdtutorials.com/esp32-cam-static-fixed-ip-address-arduino/>>. Acesso em: 03/04/2023.
- [3] Rui Santos. Configurar o ESP32 como Acess Point. Acesso em: <https://randomnerdtutorials.com/esp32-cam-access-point-ap-web-server/>>. Acesso em: 03/04/2023.
- [4] Rui Santos. Mudar as configurações da câmera do ESPre. Disponível em: <https://randomnerdtutorials.com/esp32-cam-ov2640-camera-settings/>>. Acesso em: 03/04/2023.

Dúvidas?

Módulo de Internet das Coisas



UNIVERSIDADE
ESTADUAL DO CEARÁ



Instituto Iracema
PESQUISA E INOVAÇÃO



MINISTÉRIO DA
CIÊNCIA, TECNOLOGIA
E INOVAÇÃO

